BouncyCastle FIPS KeyStore (BCFKS) Setup

CBOM Secure · Keystore Sensor (Discover_Keystore)

Overview

This guide configures the CBOM Secure Discover_Keystore sensor to discover and inventory cryptographic material in BouncyCastle FIPS KeyStore (BCFKS) files. BCFKS is a FIPS 140-2 compliant format that uses AES-256 CCM encryption internally, required wherever FIPS mode is enforced on the JVM. The sensor connects read-only and:

- Locates .bcfks files across configured filesystem paths
- Enumerates private keys, public certificates, and secret-key entries
- Records algorithm, key length, validity period, and subject/issuer metadata
- Reports assets to CBOM Secure without decrypting or exporting key material

Prerequisites

- A Java runtime (JRE or JDK) version 8 or later on the target host.
- The BouncyCastle FIPS provider JAR (bc-fips-*.jar) available — the standard bcprov-*.jar is NOT compatible.
- The keystore password(s) for all BCFKS files to be inspected.
- The CBOM sensor agent installed on the host (or one with filesystem access to the files).
- Read permission for the sensor account on the BCFKS files and the bc-fips JAR.
- Network connectivity to the CBOM platform endpoint.

Step-by-Step Guide

Step 1: Locate BCFKS Files

```
find /opt -name "*.bcfks" 2>/dev/null
# broader search:
find / -name "*.bcfks" -not -path "/proc/*" -not -path "/sys/*" 2>/dev/null
```

Confirm the sensor account can read each file; adjust group membership or permissions if needed.

Step 2: Obtain and Verify the BouncyCastle FIPS Provider JAR

The sensor requires the FIPS provider — never substitute the standard bcprov JAR. Download from the official BouncyCastle repository and verify the SHA-256 checksum:

```
# example: version 1.0.2.4 (use the latest certified release)
curl -O https://downloads.bouncycastle.org/fips-java/bc-fips-1.0.2.4.jar
mv bc-fips-1.0.2.4.jar /opt/cbom-sensor/lib/
ls -la /opt/cbom-sensor/lib/bc-fips-1.0.2.4.jar
```

Step 3: Inspect a Keystore Manually

```
keytool -list \
  -keystore /opt/app/config/keystore.bcfks \
  -storetype BCFKS \
  -storepass changeit \
  -provider org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider \
  -providerpath /opt/cbom-sensor/lib/bc-fips-1.0.2.4.jar
```

A successful listing of aliases, entry types, and creation dates confirms the sensor can perform the same inspection at scan time.

Step 4: Create a Dedicated Read-Only Service Account

```
useradd --system --no-create-home --shell /sbin/nologin cbom-sensor
chown root:cbom-sensor /opt/app/config/keystore.bcfks
chmod 640 /opt/app/config/keystore.bcfks
chown root:cbom-sensor /opt/cbom-sensor/lib/bc-fips-1.0.2.4.jar
chmod 640 /opt/cbom-sensor/lib/bc-fips-1.0.2.4.jar
```

Step 5: Configure the CBOM Secure Sensor

```
sensor:
  name: Discover_Keystore
  enabled: true
  schedule: "0 2 * * *"
connection:
  platform_endpoint: "https://cbom.example.com"
  auth_token: "cbom-token-xxxxxxxxxxxxxxxxxxxx"
  tls_verify: true
keystore:
  targets:
    - path: "/opt/app/config/keystore.bcfks"
      store_type: "BCFKS"
      store_password: "changeit"          # use a secrets reference in production
    - path: "/opt/middleware/security/app-trust.bcfks"
      store_type: "BCFKS"
      store_password: "truststorepass"
  provider_jar: "/opt/cbom-sensor/lib/bc-fips-1.0.2.4.jar"   # must be bc-fips-*.jar
  provider_class: "org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider"
  scan_paths: ["/opt", "/etc/ssl/bcfks"]
  include_patterns: ["*.bcfks"]
  inspection_timeout_seconds: 30
```

Restrict the config file permissions to protect keystore passwords:

```
chown root:cbom-sensor /etc/cbom-sensor/discover_keystore.yaml
chmod 640 /etc/cbom-sensor/discover_keystore.yaml
```

Step 6: Validate

```
systemctl restart cbom-sensor
cbom-sensor run --config /etc/cbom-sensor/discover_keystore.yaml --now
tail -f /var/log/cbom-sensor/discover_keystore.log
```

Confirm lines such as Loaded provider: BCFIPS … and Inspecting: …keystore.bcfks — 2 entries found, then verify assets under Assets → Keystores.

Common Errors

BCFKS not found (KeyStoreException)

Cause: provider_jar points at bcprov-*.jar (standard) instead of bc-fips-*.jar, which does not register the BCFKS type.

Resolution: Ensure provider_jar names a file beginning with bc-fips-; replace and restart the sensor.

mac check in GCM failed (wrong password)

Cause: The store_password does not match. BCFKS uses AES-256 CCM integrity, so a wrong password produces a MAC failure.

Resolution: Confirm the password with the provisioning team, update store_password, and restart the sensor.

Permission denied on JAR or keystore

Cause: The cbom-sensor account cannot read the provider JAR or a keystore file.

Resolution: Diagnose with sudo -u cbom-sensor ls -la …; adjust ownership/permissions per Step 4 and restart.

Security Recommendations

- Use a secrets manager for keystore passwords — never plaintext in YAML.
- Restrict the config file to root:cbom-sensor with mode 640.
- Verify the bc-fips JAR SHA-256 against the BouncyCastle website before deployment.
- Run the sensor under a dedicated, non-privileged account with no login shell.
- Use auditd to log processes opening .bcfks files and review periodically.

Conclusion

With the correct bc-fips provider JAR, a read-only service account, and accurate keystore paths and passwords, the Discover_Keystore sensor provides continuous, read-only visibility into BCFKS keystores across FIPS-regulated environments — feeding all discovered keys and certificates into CBOM Secure for expiry tracking and compliance reporting.